知识管理论坛 ISSN 2095-5472 CN11-6036/C
Knowledge Management Forum
E-mail: kmf@mail.las.ac.cn
http://www.kmf.ac.cn

【学术探索】

基于语义相似度的 CORE 论文关联关系发现及其语义服务研究

白林林 万妮
北京信息科技大学图书馆 北京 100192

摘要：[目的 / 意义] 通过对 CORE 论文关系发现过程及其服务的详细剖析，希望为我国开放获取知识库在论文内容的推荐和语义链接方面提供有力的参考和借鉴。[方法 / 过程] 从基于语义相似度的论文关联关系发现过程和基于论文关系的语义服务两方面进行分析。其中，基于语义相似度的论文关联关系发现过程包括元数据和全文内容收割、论文之间关系语义相似度计算两方面；基于发现的论文关联关系的语义服务包括论文推荐服务和关联开放数据服务。最后总结 CORE 对我国机构知识库的应用建议。[结果 / 结论] 研究发现，CORE 系统通过现有 OAI-PMH 协议自动收割开放获取知识库中的元数据，并进一步提取元数据中 URI 字段，通过 HTTP 协议下载全文。基于发现的论文语义关系提供论文推荐服务和论文关联数据服务，使得第三方系统可以利用 CORE 数据集，这些都为我国开放获取知识库（如机构知识库、开放获取期刊）在论文关系的推荐和语义链接方面提供有力的参考。

关键词：Connecting Repositories 语义相似度 论文关系 推荐系统 关联数据
分类号：G254

引用格式：白林林 , 万妮 . 基于语义相似度的 CORE 论文关联关系发现及其语义服务研究 [J/OL]. 知识管理论坛 , 2021, 6(5): 271-281[引用日期]. http://www.kmf.ac.cn/p/260/.

开放获取（open access, OA）运动推动和促进了全球科研成果的免费访问和开放获取知识库的建设与发展。但是，目前开放获取不应该只是实现科研成果的开放，而是要在现有基础上，充分发挥 OA 的潜力，通过改进现有的 OA 技术基础设施，用以支持内容的搜索、发现、挖掘、分析等这些功能。目前的大多数开放获取技术基础设施（如机构知识库、主题知识库、科研数据知识库等）大都是基于元数据的访问，而要实现开放获取内容的挖掘、分析等功能，

基金项目：本文系国家社会科学青年基金项目“基于知识图谱的领域知识结构构建方法研究”（项目编号：20CTQ007）和北京信息科技大学高教研究一般项目“大数据环境下北京信息科技大学图书馆资源利用研究”（项目编号：2019GJYB09）研究成果之一。
作者简介：白林林 (ORCID: 0000-0003-2265-7399)，馆员，博士，E-mail:bailinlin@mail.las.ac.cn；万妮 (ORCID: 0000-0003-2725-402X)，馆员，硕士。
收稿日期：2021-03-22 发表日期：2021-10-15 本文责任编辑：刘远颖

必须实现 OA 元数据集成向内容集成的有效转换。为此，由欧共体资助的项目“欧洲研究开放获取基础设施（the Open Access Infrastructure for Research in Europe, OpenAIRE）”通过建立全欧研究信息平台来收割和监测欧共体和其他国家资助者的开放获取研究成果，从而提供丰富的元数据服务和科学成果链接服务，该项目开始于 2009 年 12 月 1 日，已从第一代发展到第五代（第一代 OpenAIRE、第二代 OpenAIREplus、第三代 OpenAIRE2020、第四代 OpenAIRE-Advance、第五代 OpenAIRE-Nexus）[1]。截至 2021 年 3 月，美国的共享访问研究生态系统 (Shared Access Research Ecosystem, SHARE) 对 182 个数据源的 6 575 万多个研究成果进行了集成 [2]。法国的 HAL (Hyper Articles en Ligne) 主要对法国的科研成果进行集成，由法国国家科学研究中心的计算科学与控制研究所运行管理，目前收录了 168 个机构的 251 万多条数据 [3]。我国由 CALIS 组建的机构知识库整合系统中国高校机构知识库联盟集成了 50 家成员机构的 286 万条元数据 [4]，香港机构知识库整合系统对香港的 8 个大学的 42.6 万条数据进行了集成 [5]。但目前的这些开放获取技术基础设施，只是从元数据层面对不同来源的研究成果进行聚合和集成，并没有进一步从全文内容对论文和论文之间的关联关系进行集成和发现。CORE（COnnecting Repositories）[6] 是第一个从全文内容来发现论文之间的关联关系的系统，并将发现的论文关联关系通过不同的方式向用户提供语义服务（如推荐服务、关联数据服务）。

基于此，对 CORE 中论文关联关系的发现过程以及在此基础上提供的语义服务进行详细解析和具体应用介绍，可为我国开放获取知识库在论文内容的推荐和语义链接方面提供有力的参考和借鉴。

1 CORE 概况

CORE（COnnecting REpositories）[7] 是 2011 年由英国开放大学知识媒体研究所 P. Knoth 构建的系统 [8]，目的是通过与数字图书馆和机构知识库的紧密合作，整合分布在不同系统上的开放资源，这些资源包括英国开放获取期刊平台（Directory of Open Access Journals, DOAJ）、世界各地机构知识库和主题知识库中的元数据与全文，并在此基础上提供了一系列的资源免费访问服务来进一步促进科研成果的开放获取，这一举措对英国的开放获取运动做出了巨大的贡献，奠定了英国开放获取内容汇总的地位。因此，CORE 自创建以来就获得来自英国联合信息系统委员会（Joint Information Systems Committee, JISC）[9] 和欧盟委员会（European Commission, EC）等一系列机构的资助，并在后续通过 DiggiCORE 和 ServiceCORE 两个项目继续开发了一些平台新功能。DiggiCORE（Digging Into Connected Repositories）项目的目标是通过使用自然语言处理技术和社会网络分析方法分析大量的开放获取科研出版物，实现研究团体行为模式、研究领域趋势和研究人员引文行为的识别，以发现高影响力的论文，用于开发搜索和浏览数字馆藏更好的方法，同时形成评价科研影响力和学者影响力的新方法。ServiceCORE 项目的目标是通过进一步改进完善 CORE 技术基础设施，开发面向科研成果的主题分类系统和知识发现系统，如在 CORE Linked Data 知识库之上构建的新 Web 服务层，提供对内容和元数据的可编程访问；构建基于文本挖掘的增强型相关资源发现系统；使用文本分类技术（支持向量机）对内容进行基于主题的自动分类的工具等 [10]。

截至 2021 年 3 月，该系统已收割来自 13 799 个机构知识库和主题知识库的 2.1 亿多篇开放获取论文 [11]。CORE 系统的特点是不像其他的开放获取搜索系统只提供元数据，CORE 还集成了全文内容，确保了科研成果全文的免费访问和下载。目前，CORE 系统提供了 3 种类型的服务：原始数据访问服务、内容管理服务和内容发现服务 [12]。同时为了提高其检索率，

知识管理论坛，2021（5）:271-281
DOI: 10.13266/j.issn.2095-5472.2021.027

CORE 于 2019 实现了 CORE-MAG 映射，即将 CORE 中的数据映射为微软学术图谱 (Microsoft Academic Graph, MAG)[13]。

（1）原始数据访问服务：包括 CORE API、CORE Dataset 和 CORE FastSync 服务。CORE API 为访问 CORE 中的大量数据提供了一个入口，目前有两个版本：一个是提供 XML 或 JSON 格式数据的 RESTful API 接口，另一个是关联开放数据 SPARQL 终端 [14]。CORE Dataset 支持用户批量下载 CORE 中的数据，用于数据处理、分析和挖掘，数据包括论文元数据和全文、CORE 到 MAG 实体的映射数据。CORE FastSync 可以无缝访问从主要出版商的非标准系统中汇总的金色和混合开放获取论文，数据通过 FastSync 协议公开和共享。

（2）内容管理服务：包括 CORE Repository Dashboard 和 CORE Repository Edition 服务。CORE Repository Dashboard 是专为知识库管理员设计的知识库面板工具，目标是提供对聚合内容的管理和控制。CORE Repository Edition 是一套面向图书馆、机构知识库和内容管理商的工具套件，可用于提高机构研究成果的可发现性、数据访问的合规性等。

（3）内容发现服务：包括 CORE Recommender 和 CORE Discovery。CORE Recommender 作为一个插件，可以用于在 CORE 和其他开放获取知识库之间推荐语义相似的论文。CORE Discovery 是一个浏览器插件，支持绕过出版商免费访问 CORE 中的论文。

2 基于语义相似度的 CORE 论文关系发现过程

基于语义相似度的 CORE 论文关系发现过程包括数据获取和论文关联关系发现两个阶段。数据获取主要是通过从可用的开放获取知识库中收割元数据记录和全文内容，并对收割到的元数据和全文进行索引；论文关联关系发现主要是通过文本挖掘技术对收割到的论文之间的语义关系进行计算与发现。

2.1 CORE 数据获取

2.1.1 元数据的收割

元数据收割的来源包括开放获取知识库（机构知识库、主题知识库）和出版商数据库两类。

（1）开放获取知识库中的元数据。开放获取知识库中的元数据收割是通过开放存档倡议的元数据收割协议（Open Archives Initiative Protocol for Metadata Harvesting，OAI-PMH）[15] 请求来实现的，OAI-PMH 成功的请求返回一个 XML 文档，其中包含有关存储在知识库中的论文的元数据信息。元数据收割过程中使用到的技术是 OCLC OAIHarvester2[16]，这是一个通过 OAI-PMH 协议进行元数据收割的 JAVA 类集合工具包 [17]。

（2）出版商数据库中的元数据。针对不支持 OAI-PMH 协议的出版商数据库中的元数据，CORE 团队开发了 CORE Publisher Connector 引擎，可以无缝访问和获取出版商数据库中的金色和混合开放获取类型的论文，通过资源同步协议 FastSync 进行同步。与只收割元数据提供互操作性的 OAI-PMH 协议相比，FastSync 协议可以共享任何类型的资源（包括元数据和实际数据），并在网上提供先进的同步机制。FastSync 协议是 ResourceSync 协议的改进版本，ResourceSync 协议开始于 2011 年底，是美国国家信息标准组织（National Information Standards Organization,NISO）和开放存档倡议团队（OAI）合作开发的项目，由斯隆基金会资助，建立在同步元数据的 OAI-PMH 策略基础上，这个项目旨在加强现代网络技术的使用规范。CORE 是最早部署 ResourceSync 协议以分发大量学术文献的公司之一，这些文献可以扩展到数百万条，并且能够进行实时更新。目前，已收割了 Elsevier、Springer Nature、Frontiers 和 PLoS 四大出版商中 180 万篇开放获取的论文 [18]。

2.1.2 全文内容的下载

开放获取知识库将全文文档的 URL 作为元数据的一部分提供，因此，全文内容的获取是在从收割到的元数据中提取 URI 字段之后，通

过 HTTP 协议自动从知识库下载获得的。从开放获取知识库下载 PDF 全文是通过一组 Java 类（如 DownloadPdf 类等）来实现的，在下载的过程中为了解决下载速度慢的问题，CORE 通过使用缓冲流（BufferedStream）[19] 将全文内容先下载到服务器上，用以解决远程服务器响应非常慢时会自动取消下载的问题。

2.1.3 元数据和全文索引

完成元数据和全文收割之后，CORE 通过 Apache Lucene 对收割到的元数据和全文文档建立索引。Apache Lucene 项目开发了 3 个开源搜索软件，包括：Lucene Core、Solr、PyLucene。Lucene Core 是其核心子项目，提供基于 Java 的索引和搜索技术、拼写检查、命中突出显示和高级分析 / 标记化功能；Solr 是一个使用 Lucene Core 构建的高性能搜索服务器，具有 XML / HTTP 和 JSON / Python / Ruby 应用程序接口，支持命中突出显示、分面搜索、缓存、复制和 Web 管理界面；PyLucene 是 Lucene Core 项目的 Python 端口 [20]。

2.2 基于语义相似度的 CORE 论文关系发现

CORE 论文关联关系的发现是通过语义关系分析器组件来完成的 [21]。该组件通过自然语言处理技术从下载的论文中提取文本，然后通过计算“论文对”之间的语义相似度来识别其关联强度。为了识别和计算论文之间的语义相似性，CORE 系统通过向量空间来表示文档内容，即将内容转换为一组词语向量，并通过找到相似的向量来找到相似的文档。该系统选择使用 Apache Tika（PDFBox）工具包 [22] 从 PDF 文档中提取文本，该工具包可从 1 000 多种不同的文件类型（如 PPT、XLS 和 PDF）中识别和提取元数据和文本，并基于 TF-IDF 向量之间的余弦相似度来计算论文之间的相似度。

具体而言，可将 CORE 论文关系发现过程分为以下 4 个步骤：

（1）分词处理。对 CORE 下载到的论文进行词法分析，构建词语字典 $T=\{t_1,t_2,\ldots,t_M\}$。所有的论文可被表示为 N × M 的词语矩阵。其中，N 表示论文数量，M 表示对每篇文章进行词法分析后形成的词语数量，每篇论文对应于矩阵中某一行的向量。

（2）TF-IDF 值的计算。TF-IDF（terms frequency-inverse document frequency） 是指 TF*IDF，用来评估某个词语在文档集合中的重要程度。TF 即词频（terms frequency），指某个词语在单个文章中的出现次数；IDF 即逆文档频率（inverse document frequency） = $\log_2(N/DF)$，其中 DF(document frequency) 表示包含某个词语的文档数量。TF-IDF 的主要思想是：一个词语对一篇文章的重要性主要是依靠它在文件中出现的次数，如果这个词语在这篇文章中的出现次数越高，则表明这个词语对于这篇文章的重要性越高；同时，它还与这个词语在整个文档中出现的文章篇数有关，随着出现的篇数越多，则会降低这个词语在这篇文章中的重要性，若包含某此项的文档越少，IDF 就越大，则该词语对不同类别文档的区分度就越高。

算法流程如下：首先对文档进行分词，并去除停用词；然后统计各个词语在单个文档中出现的次数和文档集合中词语出现的次数；最后计算得出其 TF-IDF 值。

● TF 词频的计算公式如下所示：

词频（TF）= 某个词语在文章中的出现次数　公式（1）

由于需要考虑不同的文章，长度不同，需要将词频进行归一化处理，如公式（2）所示：

词频（TF）= 某个词语在文章中的出现次数 / 文章的总词数　公式（2）

● IDF 的计算公式如下所示：

逆文档频率（IDF）=$\log_2$（文档总数 / 包含该词的文档数）　公式（3）

计算逆文档频率的原因是为了去除哪些经常出现的词语，比如说“的”“我们”“他”等这类的词语，这些词语对于整篇文档重要性不高、但是出现的频率会比较多，有可能会影响到最后的计算结果，如果是经常出现的词语则不能作为文章的关键词。

知识管理论坛，2021（5）:271-281
DOI: 10.13266/j.issn.2095-5472.2021.027

●计算 TF-IDF 的值，计算公式如下所示：

TF-IDF = 词频(TF)* 逆文档频率(IDF)公式(4)

（3）排序。对文章词语的 TF-IDF 值进行排序，从中选择提取 TF-IDF 值比较大的词语，合并成一个集合，计算每篇文章对于这个集合中的词的词频，生成文章各自的词频向量，接下来计算文章词频向量之间的相似度。

（4）相似度计算。目前存在许多用于计算两个向量之间的相似性的计算方法，例如余弦相似性、dice 系数或 Jaccard 方法，并且有一些研究在计算相似性之前采用降低矢量的维数算法来提高性能。CORE 采用了最标准的相似度计算方法：在 TF-IDF 向量基础上计算余弦相似度。与其他相似度计算方法相比，TF-IDF 向量的余弦相似度方法比较成熟，已被用于自动链接生成系统中 [23]，完整性的公式如下：

$$sim(\vec{x},\vec{y}) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}\vec{y}}{|x||y|}$$ 公式（5）

可以通过夹角的大小，来判断向量的相似程度。夹角越小，余弦值越大，就代表越相似。

3 基于发现的 CORE 论文语义关系的服务

CORE 在基于发现的论文语义关系基础上为用户提供了相似论文推荐服务和关联开放数据服务。其中，相似论文推荐服务以 CORE Recommender 插件和 CORE API 形式提供；关联开放数据服务是指 CORE 将论文之间相似性的数据作为关联数据发布，并在 Linked Data Cloud[24] 中注册。

3.1 CORE 推荐服务

在 2013 年 4 月，CORE 首次发布了适用于 Eprints 知识库中的推荐系统，名称为 CORE Widget，发布在 Eprints 商店（Eprints Bazaar）中 [25]，一个用于安装 Eprints 附件组件、补丁的商店。2016 年 10 月，CORE 推出了新的版本，对原有“CORE Widget”推荐系统进行了许多改进与升级，重新命名为 CORE Recommender，新升级的推荐系统不仅支持在 CORE 中推荐相似的论文，而且也可以部署在其他知识库和期刊系统中来推荐相似论文。其中 Eprints 知识库只需在 Eprints Bazaar 中下载即可；对于其他知识库（Dspace、Fedora、OJS），只需通过插入一段 Javascript 代码片段就可安装 [26]。目前已被用于多个知识库中，如斯特拉斯克莱德机构知识库 Strathprints[27]、拉丁美洲机构知识库联合网络 LA Referencia[28]、俄罗斯国立职业师范大学机构知识库 [29]、预印本知识库 arXiv[30] 等。

为了提高所推荐的相似论文的质量，CORE Recommender 采用多个过滤器和众包机制来筛选推荐的论文，如只提供开放获取的论文、仅包含至少一组最小元数据属性的论文、包含缩略图的论文等。另外在某些情况下，CORE Recommender 可能会提供不相关的甚至错误的推荐，为此 CORE 通过为用户提供反馈按钮进行错误上报。如果用户反馈所推荐的论文不合适，CORE 会将这些论文列入黑名单，不会再在推荐列表中显示（见图 1）。

CORE Recommender 有两种使用方式。第一种方式是作为推荐系统部署在CORE系统内，向当前被访问的论文推荐相似的论文(见图 1)。第二种方式是作为推荐插件安装并集成到知识库系统或期刊系统中，当用户访问知识库中的一个论文页面时，插件会向 CORE 发送有关所访问条目的信息，CORE 会返回相似论文列表，目前提供了两种形式的相似列表：一种是来源于 CORE 知识库中的相似论文；另一种是用户访问的知识库中的相似论文（见图 2）。

3.2 CORE 论文关系关联数据服务

2011 年，CORE 发布了在 40 多万篇全文论文关系相似度计算基础上生成的 300 多万个 RDF 三元组，实现了论文之间相似度元数据的关联数据发布，以便于第三方以灵活的形式进行访问。CORE 在将论文相似度关系发布为关联数据过程中，选择 Sesame[31] 平台作为三元组存储器，用于发布关联数据。接下来笔者将对 CORE 论文关系发布为关联数据的数据模型和实现机制进行阐述。

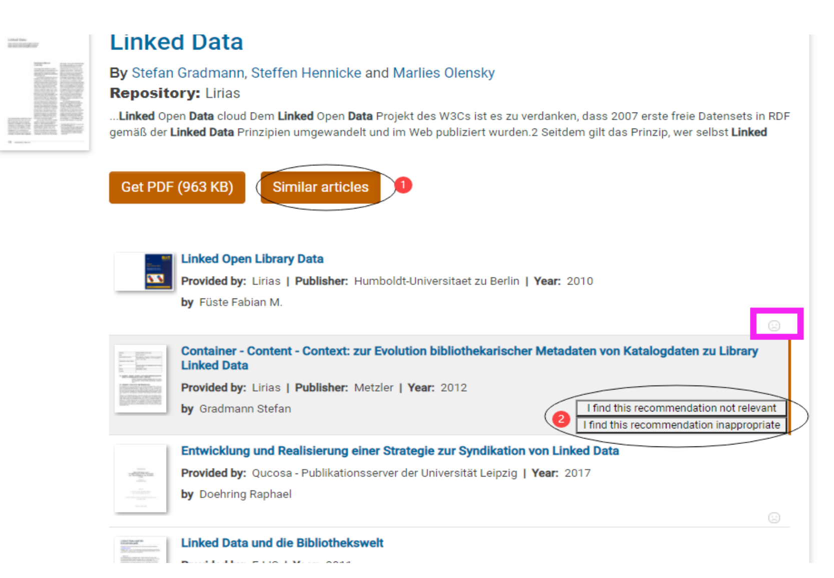


图 1 CORE 系统内提供的相似性论文

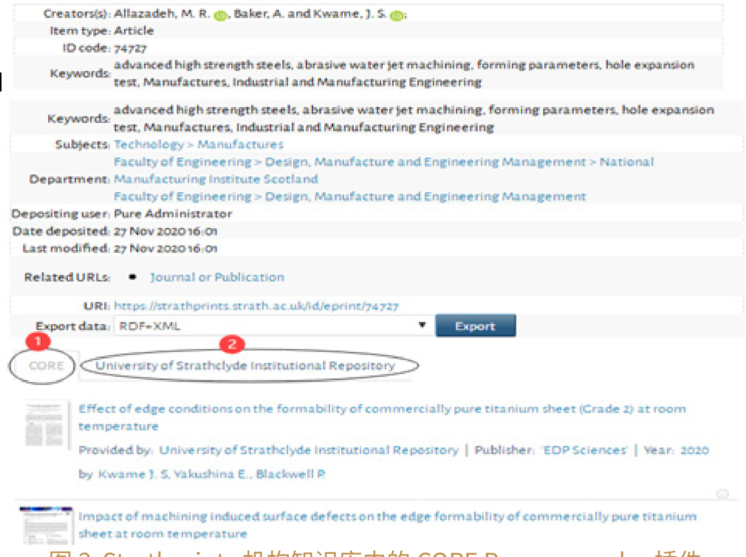


图 2 Strathprints 机构知识库中的 CORE Recommender 插件

知识管理论坛，2021（5）:271-281
DOI: 10.13266/j.issn.2095-5472.2021.027

3.2.1 CORE 数据模型

遵循关联数据原则，在将数据发布为关联数据时，尽可能复用已有的词汇表或本体来描述数据，以便外部世界更容易将新数据与已有的数据集和服务集成在一起。CORE 采用 MuSim 相似度本体（The Similarity Ontology-MuSim）[32]、书目本体（Bibliographic ontology, BIBO）[33] 以及自己构建的本体（core）来表示 CORE 知识库中论文之间的关系。

MuSim 相似度本体由伦敦大学女王玛丽数字音乐中心的 K. Jacobson、BBC 的 Y. Raimond、德累斯顿技术大学 T. Gängler 等合作开发，最初在设计时主要用于表示音乐之间的相似性，但它也可以应用到其他领域来表示两个事物之间的相似性和关联性，以便于在不同的环境下进行相关性事物的推荐和发现，这个本体中包含 5 个类和 13 个属性。在 CORE 中主要利用其属性实现对相似度计算方法、相似度权重进行语义描述。

BIBO 书目本体是由 F. Giasson 和 B. D' Arcus 合作开发的一个本体，用于在语义网中用于描述书目参考文献和引文的一些基本的类和属性，可扩展性比较强，其他的词汇也可以混合在 BIBO 本体中，如 FOAF 词汇、DC 词汇、Event 词汇等。在 CORE 中利用 BIBO 中的类和属性对论文的文献类型、作者等进行语义描述。

CORE 发布的论文相似度关系关联数据以一篇文档为主语，文档类型（rdf:type）、相似的论文 (MuSim:element)、OAI 标识符（core:hasOAIRepositoryIdentifier、core:hasOAIIdentifier）、论文之间的相似度权计算方法（MuSim:method）、相似度权重（MuSim:weight）为属性（见图 3 和图 4）。

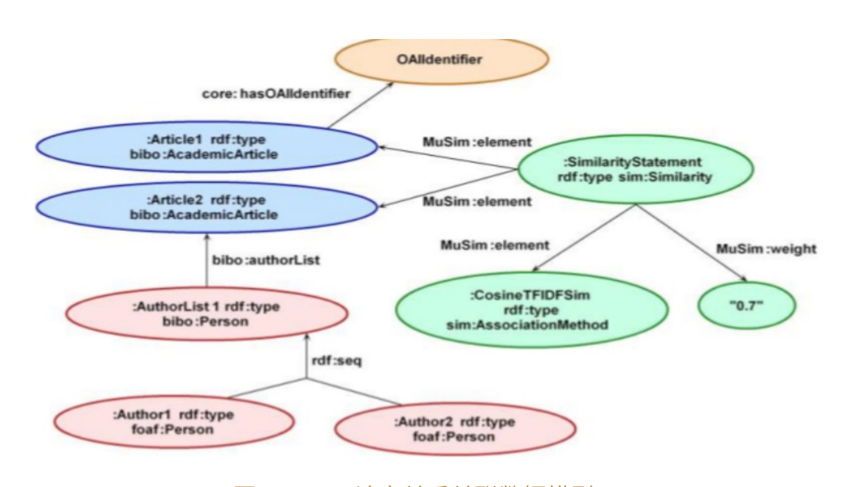


图 3 CORE 论文关系关联数据模型

3.2.2 Sesame 关联数据实现机制

Sesame 是一个查询和分析 RDF 数据的开源框架，最初由荷兰软件公司 Aduna 创建，2016 年 5 月由 Eclipse RDF4J[34] 项目继承，主要以两个 Java Web 应用程序的形式运行：OpenRDF Sesame 服务器（OpenRDF Sesame Server）和 OpenRDF 工作平台（OpenRDF Workbench）[35]。OpenRDF Sesame 服务器通过 HTTP 来访问 Sesame 库，除了提供一些服务器日志信息的查看功能外，不提供任何面向用户的功能。OpenRDF Workbench 通过一个网页界面提供面向用户的查询、浏览、更新、输出等功能。CORE 自创建以来，一直使用 Tomcat Web 服务器 [36] 作为应用程序容器，这是一个支持 Java Servlets 和 JSP 技术的 Web 服务器，所以 CORE 将 Sesame 的两个组成部分 OpenRDF Sesame Server 和 OpenRDF Workbench 部署为 Tomcat Web 服务器上的 Java Servlet 应用程序 [37]。

```xml
<rdf:RDF xmlns:ns1="http://core.kmi.open.ac.uk-sesame.open.ac.uk/openrdf-sesame/repositories/core/OAIRepositoryIdentifier" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syn
  <rdf:Description rdf:about="http://oro.open.ac.uk/62/1/JVGR_paper_2002.pdf">
    <ch_type xmlns="http://www.w3.org/TR/rdf-schema/#" rdf:resource="http://purl.org/ontology/bibo/Document"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/626800"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/629110"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/639090"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/639091"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/639092"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/639093"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/639094"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/639102"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/639209"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/639290"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/641119"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/641314"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/642250"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/642676"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/642826"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/644805"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/659022"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/659783"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/663341"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/675734"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/675784"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/677182"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/685005"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/690627"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/696825"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/713811"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/713815"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/713824"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/716688"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/718957"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/743434"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/743457"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/743506"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/778240"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/781396"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/836710"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/856184"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/862142"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/889135"/>
    <element xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/data/simNode/889802"/>
    <hasOAIIdentifier xmlns="http://core.kmi.open.ac.uk/data/">oai:open.ac.uk.OAI2:62</hasOAIIdentifier>
    <hasOAIRepositoryIdentifier xmlns="http://core.kmi.open.ac.uk/data/" rdf:resource="http://oai.rbexplorer.com/id/Open Research Online (ORO)/oai:open.ac.uk.OAI2:62"/>
    <rdf:type rdf:resource="http://purl.org/ontology/bibo/Document"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://core.kmi.open.ac.uk/data/simNode/626800">
    <method xmlns="http://purl.org/ontology/similarity/" rdf:resource="http://core.kmi.open.ac.uk/onto/method/cosine"/>
    <weight xmlns="http://purl.org/ontology/similarity/" rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.115159</weight>
  </rdf:Description>
</rdf:RDF>
```


图 4 CORE 论文关系 RDF 数据（NT）

具体而言 Sesame 分为以下 3 个层级：

（1）存储层和推理层。Sesame 的存储和推理功能通过 SAIL (Storage and Inference Layer, SAIL) API [38] 实现，这是一个从底层存储库抽象出的 API，支持内存三元组存储（in-memory triplestore）、磁盘三元组存储（on-disk triplestore）和关系型数据库存储，并有两个单独的 Servlet 软件包在永久服务器上对这些三元组存储器进行访问管理。

（2）关联数据转换层。关联数据转换过程通过 Sesame Rio（RDF）软件包实现。Sesame Rio（RDF）软件包是由一个基于 Java 的 RDF 解析器和编写器组成的简单 API，用于输入 / 输出 RDF 数据，用户可以通过在运行应用程序时将解析器和编写器放在 Java 类路径上来轻松扩展列表。

（3）关联数据查询和访问层。通过 Sesame 的 Access API 可以访问这些功能模块，它由两个独立的部分组成：Repository API 和 Graph API。Repository API 提供对 Sesame 存储库的高级访问、例如查询、存储 RDF 文件、提取 RDF 等。Graph API 为 RDF 操作提供了更细粒度的支持，例如添加和删除单个语句以及创建直接来自代码的小型 RDF 模型。这两个 API 在功能上相互补充，并且实际上经常一起使用。Sesame 支持两种查询语言：SPARQL 和 SeRQL，也可以通过 LuceneSail 添加自由文本搜索功能。

4 CORE 对我国机构知识库的应用建议

CORE 通过集成世界各地的 OA 论文元数据和全文，提供了基于论文相似度的推荐服务和基于关联数据的语义服务，完成了 OA 元数据集成向内容集成的有效转换，提高了资源的

知识管理论坛，2021（5）:271-281
DOI: 10.13266/j.issn.2095-5472.2021.027

可见度和访问率，对传统的 OA 知识库集成系统进行了发展，对我国仍处于初级阶段的机构知识库的发展和完善具有一定的新意和借鉴意义。笔者从论文关系发现过程、论文推荐服务和关联数据服务 3 个方面总结了 CORE 系统对我国机构知识库完善的启示。

在论文关系发现方面，CORE 先收割元数据，并进一步从收割到的元数据中提取 URI 字段，之后通过 HTTP 协议自动从知识库下载全文；在此基础上通过自然语言处理技术从下载的论文中提取文本，然后通过计算“论文对”之间的语义相似度来识别其关联强度。目前，我国机构知识库整合系统已实现了元数据层面的收割，并未实现全文的获取，但在所收割的元数据字段中已包含 URI 字段，后续需要通过 URI 实现全文获取，并将获得的全文通过自然语言处理技术提取文本，计算论文对之间的相似性来识别论文关系。

在论文语义推荐服务方面，CORE 通过将其开发 CORE Recommender 插件部署在 CORE 内或者其他知识库中实现论文推荐。我国机构知识库可借鉴这种思路，研发推荐服务系统或者引进 CORE Recommender 插件部署在机构知识库中，以此来为用户推荐相似论文。

在关联数据服务方面，CORE 通过利用现有的词汇表 MuSim 相似度本体、BIBO 书目本体和 Sesame 平台对论文数据进行关联化发布，方便用户更好地进行语义链接。我国可以通过分析机构知识库的数据进行建模，尽可能复用现有的成熟的词汇表对数据进行描述，并利用开源的关联数据发布工具和平台对机构知识库中的文献资源进行语义化组织和发布，从而提高资源的可发现性和可见度。

5 CORE 论文关系发现过程及服务中遇到的问题

CORE 在论文关系发现过程及提供的相关服务中也有许多问题和挑战需要去解决，具体的解决方法如下：

（1）在全文内容下载方面，主要涉及文件下载速度和数据存储成本问题。针对下载速度问题，CORE 通过使用缓冲流（BufferedStream）将全文内容先下载到开放大学服务器上，用以解决在远程服务器相应非常慢时自动取消下载的问题。有关数据存储成本问题，鉴于 CORE 需要从许多开放获取存储库中下载数据，系统需要较大的磁盘空间，同时为了执行系统备份并允许系统快速响应，选择快速串行连 SCSI（Serial Attached SCSI, SAS）磁盘。

（2）在提取文本方面，CORE 测试了 3 个 PDF 文本提取系统：iText、Apache Tika（PDFBox）和 pdftotext，最后发现虽然 Apache Tika 的提取速度非常慢但提取到的文本质量较高。最终，通过使用 BufferedStreams 先行缓冲，设法加快提取速度。

（3）在相似度计算方面，为了能够在合理的时间内发现相关的论文，涉及大量的论文组合问题。CORE 开发了一种新的启发式方法，通过使用文档频率切割标准来减少要考虑的组合数量，并考虑到计算结果的质量问题，CORE 在 Lucene 库上开发了自己的 TextAnalyzer 和 TextFilter，用于过滤数学公式、数字和其他类型的噪声数据等。

6 结语

笔者通过对 CORE 论文元数据和全文获取过程、论文之间关系语义相似度计算的论文关系发现过程以及基于发现的论文语义关系提供的服务进行了分析，为我国在开放获取知识库论文关系发现过程、论文推荐服务和关联数据服务 3 个方面提供了有力的参考，但是 CORE 也存在下载速度慢、存储开销大、PDF 中文本提取速度慢、相似度计算准确度等问题，围绕这些问题和挑战还有待进一步深入的研究。

参考文献：

[1] Openaire-history [EB/OL]. [2021-03-01]. https://www.openaire.eu/openaire-history.

[2] SHARE [EB/OL]. [2021-02-27]. https://share.osf.io/.

[3] The open archive HAL [EB/OL]. [2021-03-01]. https://hal.archives-ouvertes.fr/.
[4] 中国高校机构知识库联盟 [EB/OL]. [2021-03-01]. http://chair.calis.edu.cn/.
[5] Hong Kong Institutional Repositories (HKIR) [EB/OL]. [2021-03-01]. https://library.tu.ac.th/tu-digital-collections/hong-kong-institutional-repositories-hkir.
[6] CORE – Aggregating the world's open access research papers [EB/OL]. [2021-03-01]. https://core.ac.uk/.
[7] COnnecting Repositories [EB/OL]. [2021-03-01]. https://en.wikipedia.org/wiki/COnnecting_REpositories.
[8] Knowledge Media Institute [EB/OL]. [2021-03-01]. https://news.kmi.open.ac.uk/rostra/news.php?r=11&t=2&id=18463=KMi.
[9] CORE | Jisc [EB/OL]. [2021-03-01]. https://www.jisc.ac.uk/core#.
[10] Digging into Connected Repositories (DiggiCORE) [EB/OL]. [2021-03-01]. https://diggingintodata.org/awards/2011/project/digging-connected-repositories-diggicore.
[11] Data Providers [EB/OL]. [2021-03-01]. https://core.ac.uk/dataproviders.
[12] CORE Services [EB/OL]. [2021-03-01]. https://core.ac.uk/services.
[13] CORE Dataset [EB/OL]. [2021-03-01]. https://core.ac.uk/documentation/dataset/.
[14] Connecting Repositories (CORE) | Digging Into Data [EB/OL]. [2021-03-01]. https://diggingintodata.org/repositories/connecting-repositories-core.
[15] Open Archives Initiative Protocol for Metadata Harvesting [EB/OL]. [2021-03-01]. http://www.openarchives.org/pmh/.
[16] OAIHarvester2 [EB/OL]. [2021-03-01]. https://www.oclc.org/research/activities/oaiharvester2.html.
[17] Technical standards [EB/OL]. [2021-03-01]. https://blog.core.ac.uk/2011/03/.
[18] Releasing 1.8 million open access publications from publisher systems for text and data mining [EB/OL]. [2021-03-01]. https://blogs.lse.ac.uk/impactofsocialsciences/2018/03/22/releasing-1-8-million-open-access-publications-from-publisher-systems-for-text-and-data-mining/.
[19] Java 文 件 流 BufferedStream [EB/OL]. [2021-03-01]. https://blog.csdn.net/mariofei/article/details/51195055.
[20] Apache Lucene[EB/OL]. [2021-03-01]. http://lucene.apache.org/.
[21] KNOTH P, ROBOTKA V, ZDRAHAL Z. Connecting repositories in the open access domain using text mining and semantic data [C]// International conference on theory and practice of digital libraries :research and advanced technology for digital libraries. Berlin: Springer, 2011: 483-487.
[22] Apache Tika [EB/OL]. [2021-03-01]. https://tika.apache.org/https://tika.apache.org/.
[23] FRANCINE C, AYMAN F, THORSTEN B. Multiple similarity measures and source-pair information in story link detection[C]// Proceedings of the human language technology conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004. Boston: Association for Computational Linguistics, 2004: 313-320.
[24] CORE - Semantic Similarity of Open Access publications [EB/OL]. [2021-03-01]. https://lod-cloud.net/dataset/core.
[25] The EPrints Bazaar [EB/OL]. [2021-03-02]. https://bazaar.eprints.org/.
[26] CORE Recommender [EB/OL]. [2021-03-03]. https://core.ac.uk/services#recommender.
[27] Implementing the CORE Recommender in Strathprints: a "whitehat" improvement to promote user interaction [EB/OL]. [2021-03-03]. https://blog.core.ac.uk/2017/10/31/implementing-the-core-recommender-in-strathprints-a-whitehat-improvement-to-promote-user-interaction/.
[28] LA Referencia integrates CORE Recommender in its services [EB/OL]. [2021-03-03]. https://blog.core.ac.uk/2019/11/20/la-referencia-integrates-core-recommender-in-its-services/.
[29] CORE Recommender installation for DSpace [EB/OL]. [2021-03-03]. https://blog.core.ac.uk/2020/03/12/core-recommender-installation-for-dspace/.
[30] CORE Recommender now supports article discovery on arXiv [EB/OL]. [2021-03-03]. https://blog.arxiv.org/2020/10/15/core-recommender-now-supports-article-discovery-on-arxiv/.
[31] Sesame (framework) – Wikipedia [EB/OL]. [2021-03-06]. https://en.wikipedia.org/wiki/Sesame_(framework).
[32] The Similarity Ontology [EB/OL]. [2021-03-04]. http://grasstunes.net/ontology/similarity/0.2/musim.html.
[33] D'ARCUS B, GIASSON F. Bibliographic ontology specification [EB/OL]. [2021-03-05]. http://bibliontology.com/.
[34] Eclipse RDF4J – a Java framework for RDF [EB/OL].

知识管理论坛，2021（5）:271-281
DOI: 10.13266/j.issn.2095-5472.2021.027

[2021-03-10]. http://rdf4j.org/.
[35] Overview (OpenRDF Sesame 4.1.2 API) [EB/OL]. [2021-03-15]. http://archive.rdf4j.org/javadoc/sesame-4.1.2/.
[36] Apache Tomcat® [EB/OL]. [2021-03-15]. http://tomcat.apache.org/.
[37] Chapter1.Introduction: what is Sesame? [EB/OL]. [2021-03-17]. https://poc.vl-e.nl/distribution/manual/sesame-1.2.3/ch01.html.
[38] The SAIL API [EB/OL]. [2021-03-18]. http://docs.rdf4j.org/sail/.

作者贡献说明：
白林林：负责数据获取、研究提纲确定与论文撰写；
万妮：负责论文的修订。

Research on CORE Paper Association Discovery and Semantic Services Based on Semantic Similarity

Bai Linlin Wan Ni
Beijing Information and Science Technology University Library, Beijing 100192

Abstract: [Purpose/significance] This paper dissects the process and services of article association discovery in Connecting Repositories, and hopes to provide powerful reference for the recommendation and semantic linking of the content of articles in Chinese open access repositories. **[Method/process]** This paper analyzed the discovery process of article association based on semantic similarity and the semantic services based on article association. The discovery process of article association based on semantic similarity included metadata and full-text content harvesting, and semantic similarity calculation of article association. The semantic service based on the discovery process of article association included the CORE recommendation service and the linked open data service. And this paper summarized the application suggestions of CORE to Chinese institutional repositories. **[Result/conclusion]** This paper finds CORE system automatically harvests the metadata of the open access repositories through the existing OAI-PMH protocol, and further extracts the URI fields from the metadata to download the full-text through the HTTP protocol. Furtherly, providing article recommendation services and services of data linked articles based on the discovery of article semantic association enables third-party systems to utilize CORE datasets, it provides a powerful reference in recommendation and semantic linking of article association for open access repositories (such as institutional repositories and open access journals) in China.

Keywords: Connecting Repositories semantic similarity article association recommendation system linked data